

Sistemas Operativos

(ano letivo 2017-18)

e-fólio B

Este enunciado constitui o elemento de avaliação designado por “e-fólio B” no âmbito da avaliação contínua e tem a cotação total de 5 valores. A sua resolução deve ser entregue até às 23h55 do dia 23 de maio pelos alunos que escolheram a modalidade de avaliação contínua.

A resolução deve ser entregue através de um único ficheiro compactado .zip, que:

- (i) contém os ficheiros .c que constituem o código dos programas, prontos a serem compilados;
- (ii) contém um ficheiro pdf de formato livre, com um relatório sucinto com informações complementares de modo a permitir uma fácil compreensão do trabalho realizado. É desnecessário incluir uma listagem integral do código.
- (iii) O nome do ficheiro .zip a entregar deve seguir a seguinte convenção para o seu nome,

“NumeroAluno-PrimeiroNome-Apelido-21111-efB.zip”

Por exemplo, um aluno com número 327555 e nome Paulo ... Costa, deverá dar o seguinte nome ao ficheiro, “327555-Paulo-Costa-21111-efB.zip”

O ficheiro deve ser única e exclusivamente entregue através do recurso “E-fólio B” disponibilizado na plataforma (Nota: apenas é visível para os alunos inscritos em avaliação contínua), não sendo aceites trabalhos enviados por outras vias, como por exemplo por e-mail.

Esta é uma prova de avaliação **individual** e não “um trabalho de grupo”. A sua resolução deve provir unicamente do conhecimento adquirido e trabalho original desenvolvido pelo próprio aluno. Os alunos deverão saber distinguir claramente entre discutir os conteúdos abordados na unidade curricular (permitido) e discutir a resolução específica do e-fólio (não permitido).

I

1. [5] Escreva um programa multitarefa em linguagem C padrão e segundo a norma POSIX, de nome `mtmaxv.c`, que determine o maior valor de um vetor `v[]` com `nv` elementos. Para efetuar este processamento, o programa deve cumprir as seguintes especificações,

- O programa `mtmaxv` recebe três ou quatro argumentos na linha de comandos,

```
>> ./mtmaxv nt nv nbloco semente
```

- `nt` é o número de tarefas trabalhadoras que o programa deve criar para efetuar o processamento. Este número não inclui a tarefa principal (`main`) que não efetua processamento.

- `nv` é a dimensão (ou comprimento) do vetor a pesquisar. O programa deve com recurso a memória dinâmica criar um vetor `v[]` de inteiros (tipo `int`) de dimensão `nv` e inicializá-lo com valores aleatórios entre 0 e 100 000 000 com recurso à função de biblioteca `rand()`.

- `nbloco` é a dimensão de um bloco do vetor `v[]` a pesquisar de cada vez por uma tarefa.

- `semente` é um valor >0 opcional utilizado para inicializar o gerador de números aleatórios com recurso à função de biblioteca `srand(semente)`. Se não for dado, deve ser usado o valor por defeito de 737.

- O programa deve testar se o número de argumentos dado na linha de comandos é correto e validar os argumentos para $nt \geq 1$, $N \geq 1$ e $nbloco \geq 1$.

- O programa deve utilizar pelo menos as seguintes variáveis globais,

```
/* variaveis globais */
int *v,                /* vetor a pesquisar */
    nv,                /* comprimento vetor v */
    nt,                /* numero de tarefas */
    nbloco,            /* comprimento bloco */
    maxglobal           /* maior valor encontrado */
```

- Quando criadas, as `nt` tarefas devem receber respetivamente como argumento o seu número de ordem, entre 0 e `nt-1`.

- No início do programa, a tarefa principal (`main`) deve imprimir uma mensagem do tipo “Maximo de `v[nv]` com `nt` tarefas”.

- A estratégia geral do programa a seguir é atribuir a cada tarefa um bloco de cada vez para processamento, sendo os blocos no geral atribuídos sequencialmente até ao fim do vetor `v[]`.

- Cada tarefa tem uma variável local, `maxlocal`, inicializada a zero. Após a análise de um bloco a tarefa verifica se o seu `maxlocal` foi melhorado. Se sim, então compara-o com o `maxglobal`, atualizando `maxglobal` se for maior ou igualando-o a `maxlocal` se for menor.

- Cada vez que uma tarefa atualiza maxglobal deve imprimir uma mensagem do tipo “Tarefa (%d) encontrou max=%d” onde os %d representam respetivamente o nº de ordem da tarefa e o novo valor de maxglobal.

- As tarefas trabalhadoras terminam quando já não existem blocos para pesquisar. Note-se que a dimensão do vetor v[] não é necessariamente um múltiplo inteiro da dimensão do bloco.

- Imediatamente antes de terminar, cada tarefa deve imprimir uma mensagem do tipo “Tarefa (%d) analisou %d elementos” onde os %d representam respetivamente o nº de ordem da tarefa e o total de elementos do vetor v[] que a tarefa testou.

- No final do programa, após terem terminado todas as tarefas, a tarefa principal (main) deve imprimir uma mensagem do tipo “Maximo global=valor”.

Pondere quais as funções da biblioteca pthread que vai utilizar no programa e consulte as respetivas man pages para se informar dos detalhes de funcionamento de cada uma. Pondere também cuidadosamente quais os recursos e as estruturas de dados manipuladas pelas tarefas e que requeiram exclusão mútua no seu acesso para o bom funcionamento do programa (como por exemplo a obtenção de um bloco do vetor).

- O programa deve estar identificado com um cabeçalho similar ao seguinte,

```
/*  
** UC: 21111 - Sistemas Operativos  
** e-fólio B 2017-18 (mtmaxv.c)  
**  
** Aluno: 327555 - Paulo Costa  
*/
```

Critérios de correção:

- Programa não compila ou produz avisos (warnings) com `gcc -Wall` => 0 valores.
- Código do programa não está correta e uniformemente indentado de modo a permitir a sua leitura fácil => 0 valores
- Programa não está comentado/explicado => 0 valores. Os comentários no programa elucidam questões do código locais ao comentário. A estrutura e funcionamento do programa a nível global deve ser dada no relatório. Explique (sucintamente) o como e porquê relativamente às opções que tomou para desenvolver o programa.
- Programa correcto => 5 valores.
- Programa não funciona correctamente ou não cumpre todas as especificações => de 0 a 5 valores, sendo o programa avaliado como um todo e tendo em conta a implementação das características pedidas.

Nota ética: Nunca é de mais referir que o código a apresentar como solução para este e-fólio deve ser 100% original do aluno. A probabilidade de duas pessoas que efetivamente não comunicaram entre si, apresentarem programas “quase iguais” é considerada nula. Isto é válido para qualquer par de alunos (cópia), assim como entre um aluno e qualquer outra pessoa, em particular através da Internet (cópia/plágio), onde existem inúmeras soluções e código para os mais variados problemas, em sites, fóruns, blogs, etc.

Cumpra estritamente as normas de realização individual, como se estivesse num exame com consulta, onde pode consultar a documentação mas não pode falar com ninguém.

FIM